IN THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the present application:

1-10. (Canceled)

11. (Currently amended) A method comprising:

running a plurality of tasks in a multiprocessor system that includes a plurality of processors;

scheduling the plurality of tasks using a plurality of scheduling domains by scheduling tasks on a processor independent of the identity of the processor, wherein none of the plurality of scheduling domains is bound to any one processor of the plurality of processors;

implicitly synchronizing the tasks with regard to one or moreshared resources shared by the tasks in said system by associating said tasks with the scheduling domains, wherein each of the one or moreshared resources is assigned to one of the scheduling domains;

prohibiting tasks that are each associated with a same scheduling domain from running concurrently;

allowing tasks that are each associated with different scheduling domains to run concurrently; and

changing association of a task of the plurality of tasks from a first scheduling domain to a second scheduling domain, if the task requests a shared resource assigned to the second scheduling domain.

12. (Currently amended) A system comprising:

a plurality of processors;

a memory coupled to each of the plurality of processors, the memory storing data defining a set of tasks, each task of the set of tasks being runnable on more than one of said processors, each said task being associated with one of a plurality of scheduling domains, each of the plurality of scheduling domains controlling one or more shared resources; and

a scheduler to schedule the set of tasks using a plurality of scheduling domains by scheduling tasks on a processor independent of the identity of the processor, wherein none of the plurality of scheduling domains is bound to any one processor of the plurality of processors, where the scheduler prohibits tasks that are each associated with a same scheduling domain from running concurrently but allows tasks that are each associated with a different one of the plurality of scheduling domains to run concurrently, and wherein the scheduler changes association of a task of the set of tasks from a first scheduling domain to a second scheduling domain in response to the task's request for a shared resource controlled by the second scheduling domain.

13. (Canceled).

14. (Previously presented) A system as in claim 12, wherein at least one of the set of tasks is associated with more than one scheduling domain of the plurality of scheduling domains.

15. (Canceled).

16. (Original) A system as in claim 12, wherein said scheduler includes a plurality of runnable queues, one per scheduling domain.

17-22. (Canceled)

23. (Currently amended) A process comprising:

scheduling a plurality of tasks in a multiprocessor system that includes a plurality of processors by scheduling tasks on a processor independent of the identity of the processor, wherein none of the plurality of scheduling domains is bound to any one processor of the plurality of processors;

performing implicit synchronization of [[a]]the plurality of tasks in a multiprocessor system, said implicit synchronization dividing said tasks into the scheduling domains, at least one of the scheduling domains being associated with at least two tasks of the plurality of tasks and a resource shared by the at least two tasks, and wherein tasks within a same scheduling domain are prohibited from running concurrently even if run on different processors and tasks that are each from a different scheduling domain are allowed to run concurrently; and

moving a task of the plurality of tasks from a first scheduling domain to a second scheduling domain, in response to the task's request of a resource controlled by the second scheduling domain.

24-29. (Canceled).

30. (Currently amended) A method of scheduling a plurality of processes in a multiprocessor system, the method comprising:

associating the plurality of processes with a plurality of scheduling domains wherein none of the plurality of scheduling domains is bound to any one processor in the system;

implicitly synchronizing the plurality of processes by prohibiting concurrently executing processes that are each associated with a same scheduling domain but allowing concurrently executing processes that are each associated with a different one of the plurality of scheduling domains; and

changing association of a first process of the plurality of processes from a first scheduling domain to a second scheduling domain, if the first process requests a resource associated with the second scheduling domain.

31. (Previously presented) The method of claim 30 further comprising allowing concurrently executing processes that are not associated with any one of the plurality of scheduling domains.

32. (Previously presented) The method of claim 30, wherein at least one of the plurality of processes is associated with more than one of the plurality of scheduling domains.

33. (Previously presented) The method of claim 30, wherein each of the plurality of scheduling domains is associated with a different one of a plurality of runnable queues.

34. (Currently amended) A method implemented in a multiprocessor system, the method comprising:

executing a software program that defines a plurality of tasks and assigns each of the plurality of tasks to one of a plurality of scheduling domains, wherein none of the plurality of scheduling domains is bound to any one processor in the system;

running a plurality of processes, each of the plurality of processes performing a different one of the plurality of tasks;

prohibiting concurrently executing processes performing tasks that are each assigned to a same scheduling domain;

allowing concurrently executing processes performing tasks that are each assigned to a different one of the plurality of scheduling domains; and

allowing changing assignment of at least one task from a first scheduling domain to a second scheduling domain during executing the software program, if said at least one task requests a resource assigned to the second scheduling domain.

35. (Currently amended) A processing system comprising:

a plurality of processors;

a memory coupled to each of the plurality of processors, the memory storing instructions which, when executed by one or more of the plurality of processors, cause the one or more of the plurality of processors to perform a method comprising:

executing a software program associating a plurality of tasks with a plurality of scheduling domains and assigning a plurality of resources to the plurality of scheduling domains, wherein none of the plurality of scheduling domains is bound to any one processor of the plurality of processors;

prohibiting concurrently executing processes to perform tasks that are each associated with a same scheduling domain but allowing concurrently executing processes to perform tasks that are each associated with a different one of the plurality of scheduling domains; and

changing association of a first task of the plurality of tasks from a first scheduling domain to a second scheduling domain, if a process performing the first task requests a resource assigned to the second scheduling domain.

36. (Currently amended) A computer-readable storage medium storing instructions therein which, when executed by one or more processors of a processing system, cause the one or more processors to perform a method comprising:

executing a software program that defines a plurality of tasks and assigns each of the plurality of tasks to one of a plurality of scheduling domains, wherein none of the plurality of scheduling domains is bound to any one processor in the system;

running a plurality of processes, each of the plurality of processes performing a different one of the plurality of tasks;

prohibiting concurrently executing processes performing tasks that are each assigned to a same scheduling domain;

allowing concurrently executing processes performing tasks that are each assigned to a different one of the plurality of scheduling domains; and

allowing changing assignment of at least one task from a first scheduling domain to a second scheduling domain during executing the software program, if said at least one task requests a resource assigned to the second scheduling domain.